IST 256
Lab
Week 8 – Monday, October 17, 2011

1. Understanding how to write a method

Write a method according to the following

1. There is only one parameter and it is an integer
2. The method does the following:
    - If the value of the integer is between 0 to 99, then return the message "o.k."
    - If the value of the integer is smaller then 0 or larger then 99, then return the message "Mistake!"

First write the method header line for this method. You will need to decide on a method name and what is the type of the result.

Next write the method body to do the computation described. Do you need any local variables? Don't forget to include a return statement.

2. Understanding arrays

Write a piece of a program that will declare an array called *numbers* that can have 25 elements and then has a loop that for every array index *i* will assign the value *(i * 5)* to that element of the array.

3. Understanding Array Evaluation

Assume that a program has the following statements:

```
int [ ] values = new int[5];

for (int index=0; index < values.length; index++)
{
        if ( index == 0 )
        { values [ index ] = 10;   }
        else
        { values [ index ] = values [ index – 1 ]  + 2;   }
}
```

After this loop executes, write down the numbers in the array *values*.

values[0]	
values[1]	
values[2]	
values[3]	
values[4]	

2. Write a program with exception handling for NumberFormatExceptions

In this lab, we will write a simple program that generates values that are multiples of a number. The user will give two inputs:

the number of values that they want to see (an int)
the number that is to be the multiplier (a double)

The program will use NumberFormat Exceptions to make sure that the user inputs valid numbers.

a. Here is a sketch of the user form:

```
                         Showing Multiples

      Enter the number of values (int)      |_______________|  (textfield)
      Enter the multiplier (decimal)        |_______________|  (textfield)

                  |__ Display Multiples __|   (button)

            Results:                    (label)
```

Create the Event -> Action -> actionPerformed method for the button.

b. Write the actionPerformed method for this button:

i. In the code for the button actionPerformed method, declare an integer and a double variable. Pick good variable names for these two variables.

ii. put a try/catch statement into the code. It will initially have the form:

```
try
{

}
catch (NumberFormatException e)
{
        // print out the Exception
        System.out.println(e.toString());
        // if a number error occurs, show message and return to the user
        javax.swing.JOptionPane.showMessageDialog(null,
                "Please enter valid numbers");
}
```

iii. Inside the try block, put the code to read the two textfields into the declared variables, converting the first one to an integer and the second one to a double.

We will continue to put all the button code inside the try block because we want to skip the rest of the button and return to the user to let them try again.

iv. Add code to declare an array whose size is the integer number. Then write a for loop that for every array index i will assign the value (i * multiplier) to that element of the array. (Refer to problem 2 in this assignment.)

v. Now initialize an output String variable to the empty string and write a loop that will go over every element of the array and append an array value and a space to the output String. This doesn't have to be a multi-line label. For example, if the number of elements is 5, and the multiplier is 3.0, we could create the String:

0.0 3.0 6.0 9.0 12.0 15.0

vi. Display the string in the results label.

d. Test this program.

Try several ways to type in numbers, letters or special characters that will not be either a valid integer or double. Note the message in the output pane as well as the dialog box.

Show examples which you tried and gave NumberFormatExceptions:

Examples:

Hand in your lab sheet and your program by next Monday, October 24.